A Generalized Method for the Computational Study of the Effect of Hull Bottom Shapes on Mine-Blast Loading from Detonation of an Explosive

Jerry Clarke Raju Namburu Aaron Gupta
US Army Research Laboratory
clarke@arl.army.mil raju@arl.army.mil dasgupta@arl.army.mil

Abstract

A generalized method for generating the necessary load curves for the finite element input from CTH has been developed at ARL by using the Interdisciplinary Computing Environment (ICE). While others have successfully coupled CTH with finite element codes in the past, this method accurately represents the finite elements model's geometry on the Eulerian mesh and can be applied to any code with a pressure vs. time element loading capacity. An accurate representation of the finite element model is inserted into the CTH mesh even if the model contains shell elements.

Introduction

Light combat vehicles are increasingly being subjected to risk from a variety of highly lethal antitank land mines. As a result, there is a need for modeling and understanding the interaction of mine blast products with various shapes of bottom hull configuration and the resulting loading and damage mechanisms inflicted by explosive blast and impact. This understanding is required for crew survivability of ground combat vehicles. A previous study involving the collation of data from mines against various targets including flat plates developed a correlation function, based on dimensional analysis, for the total impulse delivered by the land mine and overburden [1].
Vulnerability of tank bottom hull floor plates subjected to blasts from shallow-buried mines was studied by Norman[2] and Hoskins et al. [3], who employed an approximate energy method approach to equate the strain energy absorbed by the plate to the energy delivered by the blast based on the conservation of energy. A number of studies have been performed for general loading mechanisms from blast-soil-structure interactions. However, the mechanics of interaction and deflection of momentum away from the affected region of the vehicle hull are poorly understood at present. The objective is to develop computational tools that allow for a simulation of a

1. (U) Hanna, J. W., "An Effectiveness Evaluation of Several Types of Antitank Mines." BRL-MR-616, U.S. Army Ballistic Research Laboratory, Aberdeen Proving Ground, MD, 1952. (UNCLASSIFIED)

2. (U) Norman, R. M., "Deformation in Flat Plates Exposed to HE Mine Blast." AMSAA-TM-74, U.S. Army Materiel Systems Analysis Activity, Aberdeen Proving Ground, MD, 1970. (UNCLASSIFIED)

3. (U) Hoskin, N. E., Allan, J. W., Bailey, W. A., Lethaby, J. W., and Skidmore, I., "The Motion of Plates and Cylinders Driven at Tangential Incidence." *Fourth International Symposium on Detonation*, ACR-126, Office of Naval Research, Pasadena, CA, 1965. (UNCLASSIFIED)

Report Documentation Page

Form Approved
OMB No. 0704-0188

1. REPORT DATE **00 DEC 2004**	2. REPORT TYPE **N/A**	3. DATES COVERED **-**

4. TITLE AND SUBTITLE **A Generalized Method for the Computational Study of the Effect of Hull Bottom Shapes on Mine-Blast Loading from Detonation of an Explosive**	5a. CONTRACT NUMBER 5b. GRANT NUMBER 5c. PROGRAM ELEMENT NUMBER
6. AUTHOR(S)	5d. PROJECT NUMBER 5e. TASK NUMBER 5f. WORK UNIT NUMBER
7. PERFORMING ORGANIZATION NAME(S) AND ADDRESS(ES) **US Army Research Laboratory**	8. PERFORMING ORGANIZATION REPORT NUMBER
9. SPONSORING/MONITORING AGENCY NAME(S) AND ADDRESS(ES)	10. SPONSOR/MONITOR’S ACRONYM(S) 11. SPONSOR/MONITOR’S REPORT NUMBER(S)

12. DISTRIBUTION/AVAILABILITY STATEMENT
Approved for public release, distribution unlimited

13. SUPPLEMENTARY NOTES
See also ADM001736, Proceedings for the Army Science Conference (24th) Held on 29 November - 2 December 2005 in Orlando, Florida., The original document contains color images.

14. ABSTRACT

15. SUBJECT TERMS

16. SECURITY CLASSIFICATION OF:			17. LIMITATION OF ABSTRACT	18. NUMBER OF PAGES	19a. NAME OF RESPONSIBLE PERSON
a. REPORT **unclassified**	b. ABSTRACT **unclassified**	c. THIS PAGE **unclassified**	**UU**	**6**	

Standard Form 298 (Rev. 8-98)
Prescribed by ANSI Std Z39-18

mine blast load on various shapes of the lower hull bottom. The model outputs are evaluated to characterize the blast-structure interaction and the influence of the shape of the hull bottom in minimizing the impulsive load. This study will improve our understandings of the influence of blast loading functions and the transfer of the load data to a nonlinear finite element model of the lower hull to obtain transient structural response.

In the past, to perform such a computational study, CTH (a finite volume, shock physics code) has been coupled with different Lagrangian finite element codes like Pronto3D and LS-Dyna, to solve blast-structure interaction problems. In many situations, a two-way coupling of these codes is unnecessary. Specifically, when the deformation of the structure has little impact on the developing blast, a one-way coupling is sufficient. Unfortunately, when the structure is complex, and particularly when the model contains shell elements, accurately generating the load curves for the finite element input can be difficult.

A generalized method for generating the necessary load curves for the finite element input from CTH has been developed at ARL by using the Interdisciplinary Computing Environment (ICE)[4]. While others have successfully coupled CTH with finite element codes in the past, this method accurately represents the finite elements model's geometry on the Eulerian mesh and can be applied to any code with a pressure vs. time element loading capacity. An accurate representation of the finite element model is inserted into the CTH mesh even if the model contains shell elements. Using this method, an example problem of a land mine interacting with a complex vehicle structure is presented.

A New Approach

The requirement of an end-to-end simulation capability for weapon-target interaction applications revealed the need for a seamless interdisciplinary capability. An interdisciplinary capability can be addressed either by developing coupled software from scratch or by integrating existing legacy software. The first approach involves developing and testing the entire software spectrum, which is time consuming and may require more time to reach the end user. The second approach involves coupling existing software from individual disciplines, which takes advantage of the tremendous "stove-pipe" developments already made in computer science and general computational sciences. This latter approach requires data management, seamless data movement, and robust modular scalable algorithms, and is the central theme of this achievement. To address this problem, a research effort entitled, "Interdisciplinary Computing Environment for Weapon-Target Interaction" was undertaken to produce an entirely new capability.

---

[4] . (U) Clarke, J.A., Namburu, R.R, "A distributed computing environment for interdisciplinary applications" Concurrency and Computation: Practice and Experience, Volume 14 No.13-15 Nov.-Dec. 2002, pp 1161-1174

The uniqueness and significance of the research contribution in the newly developed Interdisciplinary Computing Environment approach is the development of a common data hub, which is both a data model and data format. That means the information about the data values and "how the data are used" are available. Known as the eXtensible Data Model and Format (XDMF), the data hub utilizes XML and hierarchical data format version 5 (HDF5) to provide a flexible yet powerful active data hub, as shown in figure 1. The transfer of data is handled by a distributed shared-memory system called Network Distributed Global Memory (NDGM), which provides access to a virtual, contiguous buffer through a client-server architecture. A widely used HDF5 is used to provide an NDGM buffer with a structure. The common data hub facility provided by HDF5 and NDGM is effectively used to manage data between different software systems and to coordinate activities between different codes. This enables researchers and engineers to quickly couple production level parallel high-performance computing codes from different disciplines and ultimately develop coupled algorithms and approaches for addressing both one-way coupled and fully coupled weapon-target interaction applications in a seamless way.

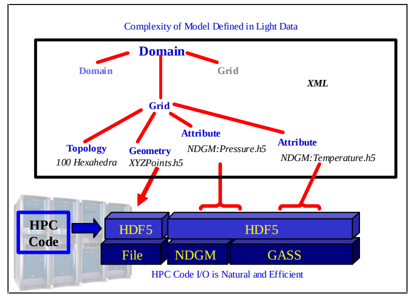

Figure 1. XDMF.

One Way Coupling

To address the requirement of one-way coupled land mine-vehicle structure interaction, this capability has been used to couple the finite-element code LS-Dyna* (a commercial Lagrandian software) with the finite volume shock physics code CTH (Department of Energy Eulerian software). This allows the strengths of both codes to be utilized to produce a result that is not possible by either code separately. CTH is widely used for modeling the dynamic loading on a structure as a result of the detonation of an explosive, while LS-Dyna is an appropriate tool for simulating the large deformation of a thin structure, such as a vehicle hull.

This type of one-way coupling is appropriate for this particular application because the deformation of the structure occurs at a rate such that it has little effect on the developing blast, but the blast has a significant impact on the deformation of the structure. This approach was automated and used by the U.S. Army Research Laboratory (ARL)

---

* LS-Dyna is a registered trademark of Livermore Software Technology.

Weapons and Materials Research Directorate researchers to complete these analyses and not only improved designs and understanding of the mine blast-structure interaction, but also significantly reduced the analysis cycle time.

Previous efforts have attacked the specific problem of one-way coupling the Eulerian shock physics code CTH with Lagrangian finite-element structural response codes. Typically, this involves the use of “tracer points,” which is a specific feature of the Eulerian code and difficult to implement in the general sense. In addition, these methods did not address the problem of accurately representing complex geometries on the structured Eulerian mesh.

In addition to the scalable coupling methods, one of the technical challenges was to use consistent meshes for both software packages. The computational domains modeled by individual codes are entirely different and not easily coupled together. Note that LS-Dyna is an unstructured finite-element approach and CTH is a finite volume-based structured mesh approach. Typically, armored vehicle hull structures are geometrically complex and are represented using unstructured thin “shell elements”. LS-Dyna uses shell elements to represent thin structures. These elements have no thickness, but accurately capture the deformations of thin structural members at a much lower computational cost than using many small hexahedral elements through the thickness of the structure. CTH, being a finite volume code based on a structured mesh, has no concept of shell elements. Introducing a structure composed of these elements into a structured mesh presents a technical challenge.

Preparing Geometry

As shown in figure 2, the method used to accomplish this task starts with giving these shell elements a thickness so they can be introduced into a structured mesh while still maintaining the shape of the thin structure. This is done by extruding each quadrilateral shell element in the direction opposite its “normal” direction (i.e., toward the inside of the entire structure). The resulting hexahedra are then decomposed into tetrahedra and converted into a data format that CTH can use for input.

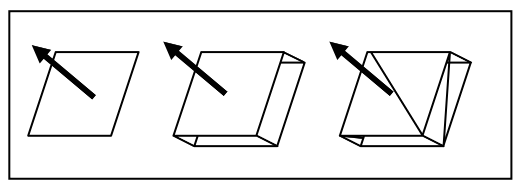

Figure 2. Shell elements are extruded to form tetrahedral elements.

But a simple extrusion is sometimes insufficient. If the shell’s normals are used, problems arise if adjoining shell normal vectors differ significantly. In this case, the back side of the extruded shell can extend beyond the surface of the adjoining shell, as depicted by the side view of the red and blue extruded shells below in figure 3. To alleviate this problem, normal vectors are

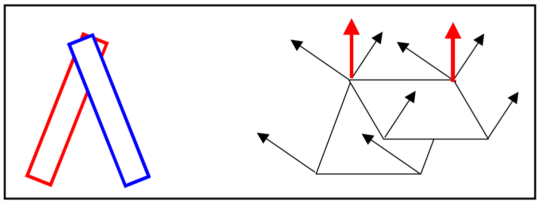

Figure 3. Normal vectors are averaged to represent the surface correctly.

calculated on the nodes and averaged. New node positions are then calculated using these vectors, and the connectivity is generated.

In addition to the newly generated geometry for material insertion, the CTH input consists of a description of the explosive charge, and potentially soil, to model the ground plane and air surroundings. CTH uses this information to simulate the explosive detonation and calculates the pressure loads as they build on the structure. The pressure information for the entire mesh is saved to the common data model and format XDMF at regular intervals. Once the load on the structure has dissipated, the CTH calculation is halted.

Extracting Pressure-Time Histories

Using the original thin shell structure, the pressure information is "probed" in order to generate a pressure-vs.-time history for each of the shell elements. That is, data are mapped, using interpolation, from the structured mesh onto points at the center of the shell faces. This information can then be input into the LS-Dyna calculation to simulate the structural deformation that will result from the explosion impinging on the vehicle structure. Sometimes the probed pressure observed directly on the surface can oscillate in the calculation. It is usually desirable to probe the pressure vales just off of the surface. For this reason, the pressure values are sampled a small distance in the direction of the shell normal vector, placing them into the blast field and off of the surface. This is depicted in figure 4 below.

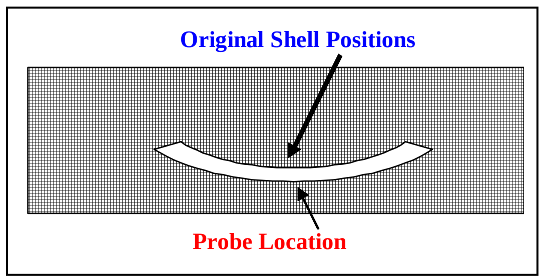

Figure 4. Pressures are probed slightly off of the original surface.

An Example

As a test case, a complex vehicle hull shape was taken from an LS-Dyna input file and converted using the previously described method. Shown in figure 5 is the original structure with an isosurface of the blast from CTH.

Since there were experimental results available for this configuration, the results of the entire simulation can be compared. To test the coupling method itself, results were compared with simple geometries that could be accomplished with the tracer point methods described earlier. Validation of the entire

simulation will require validation of many factors, including the blast and material models in both CTH and LS-Dyna.

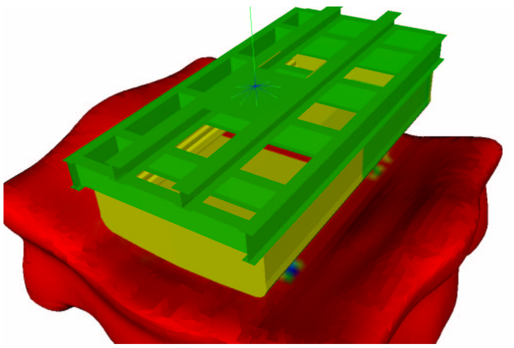

Figure 5. Mine Blast Interacting with Vehicle Hull

The process of coupling CTH to Dyna was accomplished with the development of several scripts written in the Python[5] scripting language. Python provides a convenient method to "glue" together many smaller software components. For example, XDMF access, the Visualization Toolkit,[6] and the various parsers are combined to create a single tool, without re-compiling or linking any system code.

The first script accomplishes the parsing of the original LS-Dyna input and generates an XDMF dataset of the extruded tetrahedral mesh. The next script uses the XDMF mesh as input and generates both the CTH input file and the geometry file needed for material insertion. The last script processes the CTH output and generates the pressure-vs.-time histories and formats them for input to LS-Dyna.

[5] Python home page. http://www.python.org (accessed May 18, 2004).

[6] Schroeder, W.; Martin, K.; Lorensen, B. The Visualization Toolkit 2nd Eds; Prentice Hall PTR, Upper Saddle River, NJ, 1998.

Summary

The Interdisciplinary Computing Environment (ICE) provides a new capability for the physics-based simulation of weapon-target interactions. This new capability is being used at ARL in armor/anti-armor designs and survivability of new systems against land mine threats.

Bibliography

Littlefield, D. L. *A Brief Description of New Algorithms Incorporated into CTH: A Model for Rigid Obstacles and Interface for Coupling With Structural Codes*; Texas Institute for Computational and Applied Mathematics, The University of Texas: Austin, TX, 2001.

Clarke, J. A.; Namburu, R. R. A Distributed Computing Environment for Interdisciplinary Applications. *Concurrency and Computation: Practice and Experience* **2002**, *14* (13), 1161–1174.

Clarke, J. Emulating Shared Memory to Simplify Distributed Memory Programming. *IEEE Computational Science and Engineering* **1997**, *4* (1), 55–62.